

DELETE FROM Students WHERE Major = "Undeclared";
 DELETE FROM Students WHERE Enrollment Date < "2022-01-01";
 DELETE FROM Students WHERE Age > 21;
 DELETE FROM Students WHERE First Name = 'Ethan' and Last Name = 'Taylor';
 DELETE FROM Students WHERE Age < Null;
 DELETE FROM Students WHERE LastName LIKE 'C';
 DELETE FROM Students WHERE Enrollment Date < "2022-01-01";
 DELETE FROM Students WHERE Major = "Physics";
 DELETE FROM Students;

SET 2

CREATE TABLE CUSTOMERS (
 CUSTOMERID INT PRIMARY KEY, AutoIncrement,
 First Name VARCHAR(50),
 Last Name VARCHAR(50),
 Email VARCHAR(100), unique,
 PhoneNumber VARCHAR(25),
 Address VARCHAR(200),
);

CREATE TABLE Payments (
 AccountNumber INT PRIMARY KEY,
 CUSTOMER ID INT FOREIGN KEY,
 Account Type VARCHAR(20),
 Balance Decimal(10, 2),
 Date Created (DATE)
); FOREIGN KEY (CUSTOMER ID) REFERENCES Customers (Customer ID)

INSERT INTO CUSTOMERS (Customer ID, First Name, Last Name, Email, Phone Number, Address) VALUES
 (1, 'John', 'Doe', 'john.doe@email.com', '1234567890', '123 Main St, Springfield'),
 (2, 'Jane', 'Smith', 'jane.smith@email.com', '10987654321', '456 Elm St, Townville');

(~~id~~, 'Mike', 'Johnson', 'mike.johnson@email.com', '1171 Street', '789 Oak
St, Villageville');

INSERT INTO Accounts (~~WHERE~~ (Account Number, Customer ID, Account
Type, Balance, Date created) VALUES

(1001, 1, 'Savings', '5000', '2023-01-15'),
(1002, 2, 'Checking', '1500', '2023-02-20'),
(1003, 2, 'Savings', '2000', '2023-03-01'),
(1004, 3, 'Checking', '3000', '2023-03-10');

Update ③
UPDATE ~~FROM~~ Accounts SET Balance = 5500 ~~WHERE~~ Account Number
= 1001;

UPDATE Customers SET Email = 'jane.smith@newdomain.com' ~~WHERE~~
(Customer ID = 2;

UPDATE Accounts SET Balance = Balance * 1.20 ~~WHERE~~ Account Type
'Savings';

SELECT Customers.Customer ID, Customers.First Name, Customers.Last Name,
Accounts.Balance

FROM Customers
JOIN Accounts ON Customers.Customer ID = Accounts.Customer ID
WHERE Accounts.Accounts Type: 'Savings';

SELECT Customers.Customer ID, Customers.First Name, Customers.Last Name,
Accounts.Balance

FROM Customers
JOIN Accounts ON Customers.Customer ID = Accounts.Customer ID
WHERE Accounts.Balance > 3000 AND Accounts.Account Type =
'checking';

SELECT CUSTOMER ID, Account Number, Account Type, Balance FROM Accounts WHERE Balance < 2000;

5) DELETE FROM Accounts WHERE Account Number = 1002;
DELETE FROM Customers WHERE Phone Number LIKE '1234';
DELETE FROM Accounts WHERE DateCreated < '2023-01-01';

6) SELECT Customers.First Name, Customers.Last Name, Accounts.Account Type
FROM Customers
JOIN Accounts ON Customers.Customer ID = Accounts.Customer ID
WHERE Accounts.Balance > 2000;

SELECT Account Type, SUM(Balance) AS Total Balance
FROM Accounts

WHERE Account Type = 'Saving';
GROUP BY Account Type;

SELECT Customers.First Name, Customers.Last Name, Accounts.Account Number, Account
FROM Customers LEFT JOIN Accounts ON Customers.Customer ID = Accounts.Customer ID

CREATE TABLE Library C

Book ID INT PRIMARY KEY, AUTO-INCREMENT,

TITLE VARCHAR(50),

AUTHOR VARCHAR(50),

PUBLISHER VARCHAR(50),

GENRE VARCHAR(50),

PUBLISHED YEAR INT,

ISBN VARCHAR(50), UNIQUE,

pages INT,

Copies Available INT,

PRICE DECIMAL(10,2),

);

⑦ ALTER TABLE Accounts
ADD CONSTRAINT check-balance CHECK (Balance >= 0);

ALTER TABLE Customers
MODIFY COLUMN Email VARCHAR(100) NOT NULL UNIQUE;

ALTER TABLE Accounts
ADD CONSTRAINT fk-customer FOREIGN KEY (customerID) REFERENCES
Customers (customerID);

ALTER TABLE Accounts
ADD CONSTRAINT check-account-type CHECK (AccountType IN ('Savings',
'Checking'));

⑧ SELECT Customers.FirstName, Customers.LastName, Accounts.AccountNumber,
MAX(Accounts.Balance) AS HighestBalance
FROM Customers
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID
GROUP BY Customers.CustomerID
ORDER BY HighestBalance DESC
LIMIT 1;

BEGIN TRANSACTION;

UPDATE Accounts

SET Balance = Balance - 1000

WHERE AccountNumber = 1003;

UPDATE Accounts

SET Balance = Balance + 1000

WHERE AccountNumber = 1001;

COMMIT;

SELECT Customers.FirstName, Customers.LastName, SUM(Accounts.Balance)
TotalBalance FROM Customers
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID
GROUP BY Customer.CustomerID;

9) SELECT AVG(Balance) AS AverageBalance
FROM Accounts;

SELECT COUNT(*) AS TotalSavingAccounts
FROM Accounts
WHERE AccountType = 'Saving';

10) BEGIN TRANSACTION;
UPDATE Accounts
SET Balance = Balance - 500
WHERE AccountNumber = 1002;
UPDATE Accounts
SET Balance = Balance + 500
WHERE AccountNumber = 1003;
COMMIT;